
K8s Security
На разных уровнях абстракции
Александр Сунгуров
HL
HighLoad++
2022





K8s Security

На разных уровнях абстракции



Александр Сунгуров

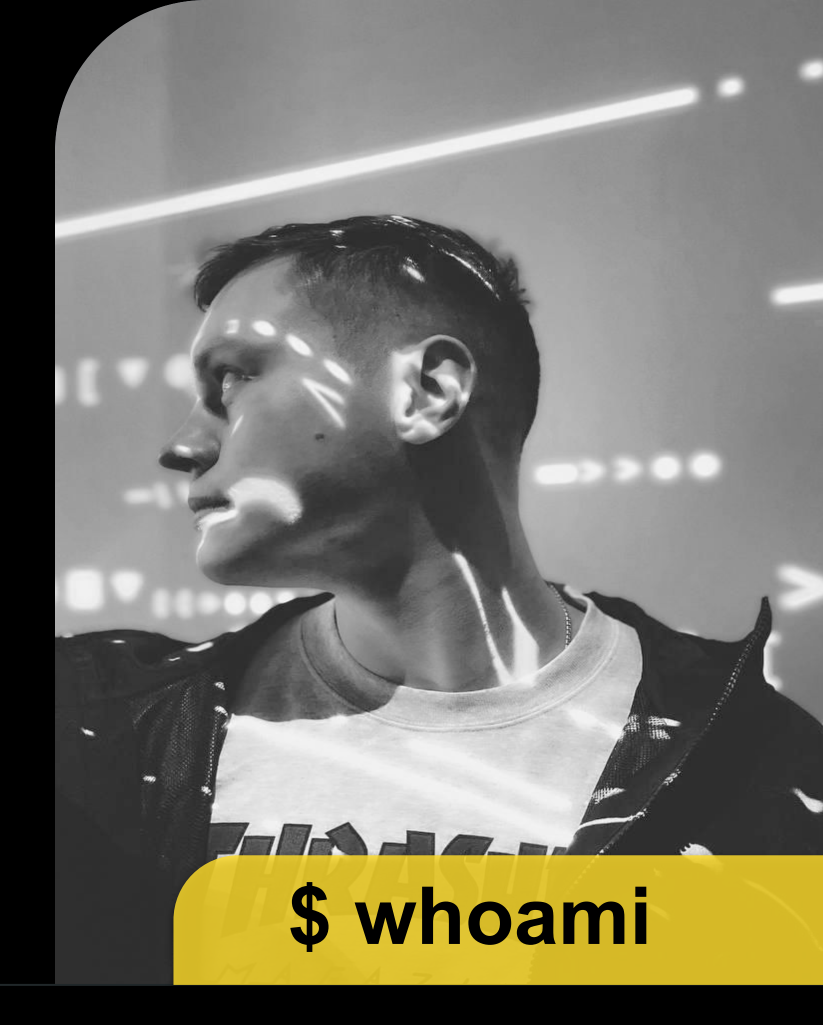
Александр Сунгуров

Архитектор по информационной безопасности

Exness

@ alexander.sungurov@exness.com

@Banzay021



K8s security

Что это?

Опрос:



Кому будет полезен доклад:



Кто активно использует K8s



У кого есть критичные сервисы/ PCI DSS в K8s



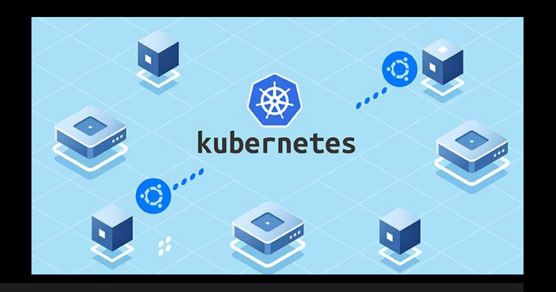
Кто использует в K8s в облаках

О чем поговорим?

- Проблемы, которые возникают
- Использование Admission Controller. PEP
- CNI, Zero trust + сетевое разделение
- Istio
- Доступ к K8s
- Безопасность среды
- Аудит и логирование
- Пример архитектуры



Безопасность хостов



- Свап и дампы должны быть отключены для K8s с критичными сервисами
- Воркеры должны быть в приватных сетях
- По умолчанию все сетевые доступы запрещены
- **TLS** по умолчанию

- Контроль целостности файлов
- Агенты **SIEM**
- Контроль уязвимостей и патч-менеджмент
- Аудит хостов

K8s Admission controllers

API request

API HTTP handler

Authentication Authorization

Mutating admission

Object Schema Validation

Validating admission

Persisted to etcd

Webhook

Webhook

Webhook

Admission Controller Phases

Open policy agent

Open Policy Agent — это механизм политик для K8s с открытым исходным кодом, который объединяет применение политики во всем стеке.
Сервис на схеме — **Istio.**

Request, Event, etc.
Service
Query
(any JSON value)
Decision
(any JSON Value)
OPA
Policy
(Rego)
Data
(JSON)

Open policy agent

OPA разделяет принятие решений по политикам от обеспечения их соблюдения.

Когда сервису нужно получить решение по политикам, он направляет запрос к OPA и предоставляет структурированные данные, например **JSON** на вход.

OPA принимает свободно структурированные данные на вход.

Open policy agent

Rego был вдохновлен Datalog, который является хорошо понятным языком запросов десятилетней давности.

Rego расширяет **Datalog** для поддержки структурированных моделей документов, таких как **JSON**.



Open policy agent

Запросы Rego — это утверждения о данных, хранящихся в OPA.

Эти запросы могут использоваться для определения политик, которые перечисляют экземпляры данных, нарушающих ожидаемое состояние системы.

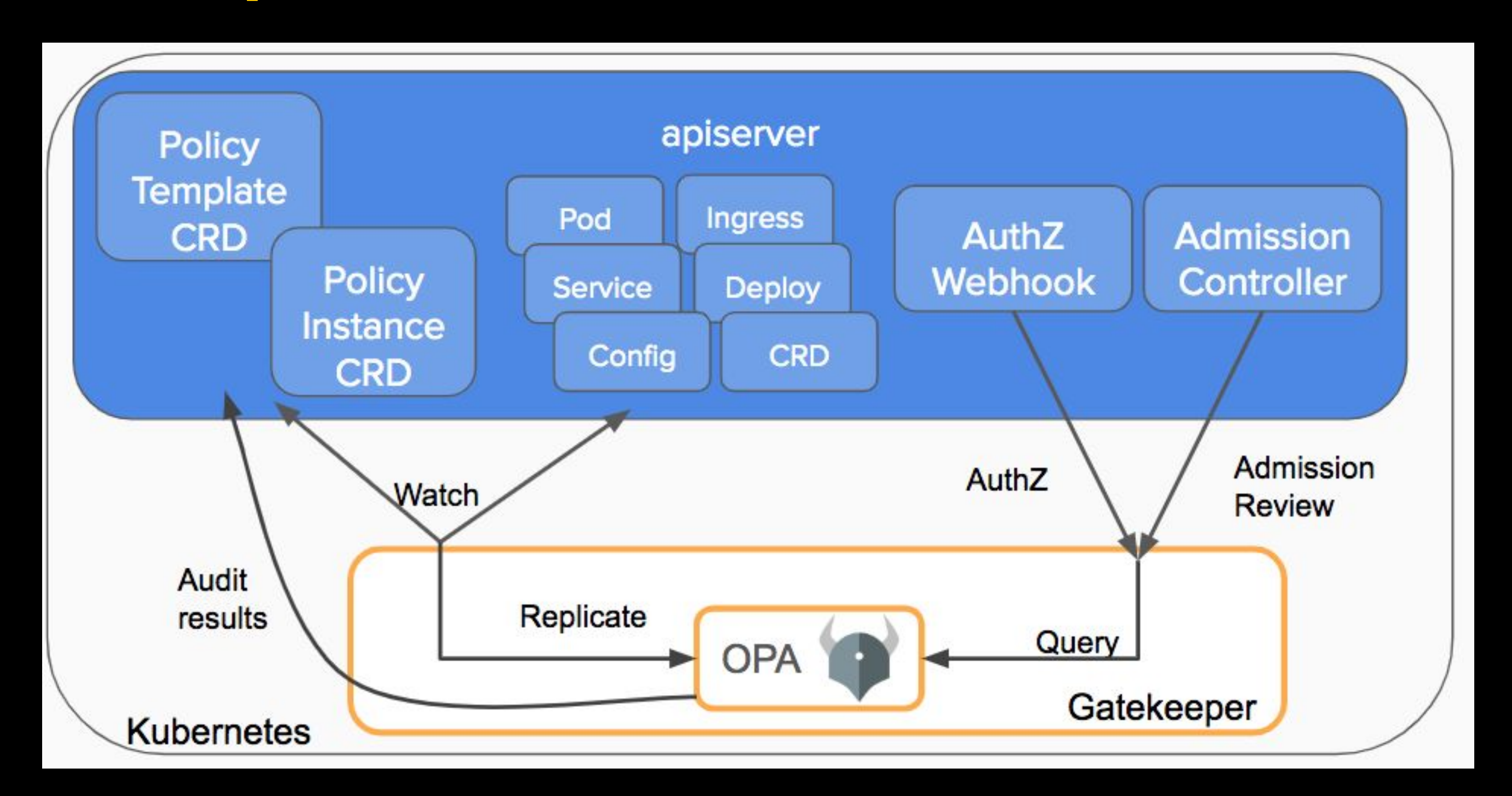
```
package authz
import future.keywords
allow if{
    input.path == ["users"]
    input.method == "POST"
}
allow if {
    input.path == ["users",
input.user_id]
    input.method == "GET"
}
```

Gatekeeper



Механизм **Gatekeeper** разработан для того, чтобы позволить администраторам обнаруживать и отклонять несоответствующие коммиты в системе **"инфраструктура как код"**, что еще больше усиливает по соблюдению требований и предотвращает публикацию уязвимых сервисов.

Gatekeeper



apiserver
Policy Template CRD
Policy Instance CRD
Pod
Ingress
Service
Deploy
Config
CRD
AuthZ Webhook
Admission Controller
Watch
AuthZ
Admission Review
Audit results
Replicate
OPA
Query
Gatekeeper
Kubernetes

Open policy agent

```
>kubectl run nginx --image nginx
Warning: [deny-latest-tag] container <nginx> does not have image tag <nginx>
Warning: [container-has-resources-defined] container <nginx> has no resource limits
Warning: [container-has-resources-defined] container <nginx> has no resource requests
Warning: [psp-deny-capabilities] container <nginx> is not dropping all required capabilities. Container must drop all of ["ALL"]. Capabilities expand some host properties.
Warning: [psp-read-only-root-filesystem] only read-only root filesystem container is allowed: nginx.
pod/nginx created
>
```

```
>kubectl apply -f nginx-secured.yaml
pod/nginx-secured created
>kubectl get po
NAME            READY   STATUS    RESTARTS   AGE
bysubox         1/1     Running   0          48d
httpd66         1/1     Running   0          56d
nginx           1/1     Running   0          5m10s
nginx-secured   1/1     Running   0          19s
```

Open policy agent

```
  input_containers[c] {
            c :=
input_object_container_spec.ephemeralContainers[_]
      }

      input_allow_privilege_container(c) {
            c.securityContext.privileged == true
      }
```

Open policy agent

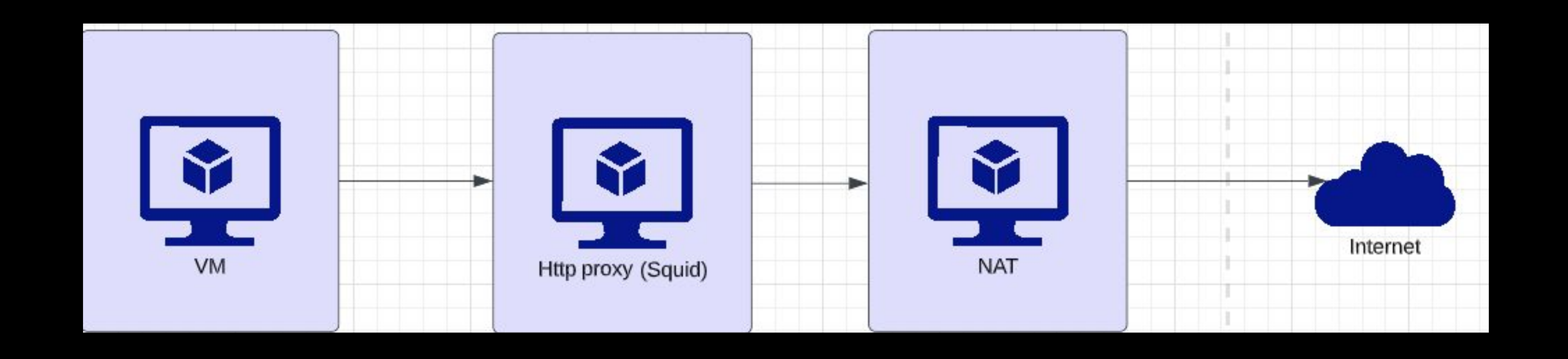
```
violation[{"msg": msg, "details": {}}] {
            c := input_containers[_]
            input_allow_privilege_container(c)
            msg := sprintf("Privileged initContainers is not allowed: %v. Running as a privileged gives full access to the host that the container is running on. This is unacceptable from a security point of view.", [c.name])
        }
```

Open policy agent

```yaml
    - name: etc
      mountPath: /var/run
    - name: var
      mountPath: /var/cache/nginx
    securityContext:
      readOnlyRootFilesystem: true
      capabilities:
        drop: ["ALL"]
  volumes:
  - name: etc
    emptyDir: {}
  - name: var
    emptyDir: {}
```

Доступ в интернет

```
acl localnet src 192.168.0.0/24
acl forbidden dstdomain "/etc/squid/forbidden_domains"
http_access allow localnet !forbidden
```



CNI

- Calico
- Cilium
- Flannel
- Weavenet





Kubelet

cri plugin

containerd

Flannel CNI plugin

Bridge CNI plugin

host-local IPAM CNI plugin

CNI

Istio: mesh



Istio расширяет возможности **Kubernetes** для создания программируемой сети с поддержкой приложений с использованием мощного прокси-сервера службы **Envoy**.

Istio: особенности

- Безопасная связь между службами в кластере с помощью шифрования **TLS**
- Автоматическая балансировка для **HTTP, gRPC, WebSocket** и **TCP traffic**
- Детальный контроль поведения трафика
- Конфигурируемый уровень политики
- Автоматические показатели, журналы и трассировки

Istio: особенности

Service A
Envoy proxy
Service B
Envoy proxy
discovery
configuration
certificates
metrics
Istio
control plane
discovery
configuration
certificates
metrics

Istio: Как это работает

- **Data plane** взаимодействует со всеми микросервисами решения
- **Service mesh** использует прокси-сервер для перехвата всего вашего сетевого трафика, предоставляя широкий набор функций, зависящих от приложений, в зависимости от заданной вами конфигурации



Istio: Как это работает

- Прокси-сервер **Envoy** развертывается вместе с каждой службой, которую вы запускаете в своем кластере, или запускается вместе со службами, запущенными на виртуальных машинах

- **Control** принимает желаемую конфигурацию и ее представление о службах и динамически программирует прокси-серверы, обновляя их по мере изменения правил или среды



Istio: Безопасность

- Центр сертификации (**CA**) для ключей и менеджмента сертификатов
- Конфигурационное API доставляет параметры до:
  - политики аутентифкации
  - политики авторизации
  - политики именования

Istio: Безопасность

- Сайдкары и прокси работают как точки распространения политик для обеспечения безопасного взаимодействия клиентов и серверов

- Набор расширений прокси-сервера **Envoy** для управления телеметрией и аудитом

Istio: Как это работает

Istio Mesh

Data plane

Service A

Service B

JWT+ TLS
mTLS

APIs
Content

Ingress

mTLS

Proxy

mTLS

HTTP, gRPC, TCP

Proxy

mTLS

Egress

JWT + TLS
mTLS

External
API

Control Plane Interface

Control plane

istiod

Certificate
authority

Authentication
policies

Network
configuration

Authorization
policies

APIserver
configuration

Key:

Data plane
traffic

Control plane
traffic

Local
authorization

Certificate

PKI

- **CA**-сертификат хранится в **HSM**
- Каждая среда K8s **со своим pki**
- Необходимо создать промежуточный **CA csr** и подписать в **CA**
- Необходимо создать промежуточный **CA csr** для кластера и подписать на промежуточным **CA**
- **Istio создает csr** и сертификаты для приложений **сам**

PKI

Root Certificate

example.com

reference

sign

Intermediate Certificate

example.com

reference

sign

Identity Certificate

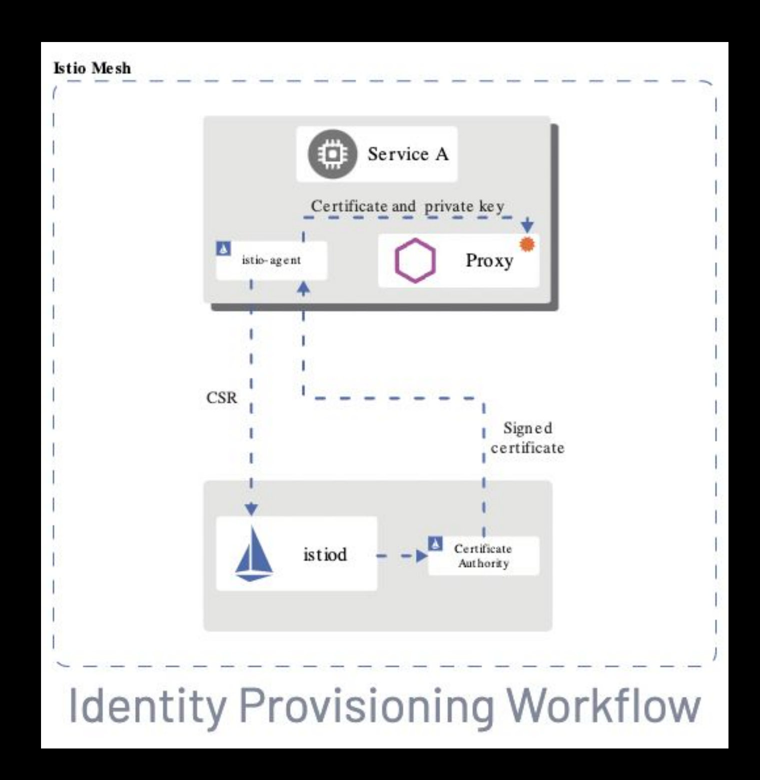
test.example.com

Istio: управление сертификатами и учетными записями

- **Istio** предоставляет учетные записи с **X.509**-сертификатами для каждого сервиса
- **Istio**-агенты, работая вместе с каждым прокси-сервером **Envoy**, взаимодействуют с **istiod**, чтобы автоматизировать масштабную ротацию ключей и сертификатов

Istio: управление сертификатами и учетными записями

Istio предоставляет автоматизированный флоу с выпуском и управлением сертификатами для сервисов.



Istio: Аутентификация

Peer authentication: используется для аутентификации от службы к службе для проверки клиента, устанавливающего соединение.

Предлагает **mutual TLS** в качестве полнофункционального решения для транспортной аутентификации, которое может быть включено без необходимости изменения кода приложения.

Istio: Аутентификация

Request authentication: Используется для аутентификации конечного пользователя для проверки учетных данных, прикрепленных к запросу.

Istio обеспечивает аутентификацию на уровне запроса с проверкой веб-токена **JSON (JWT)** и оптимизирует опыт разработчика с использованием пользовательского поставщика аутентификации или любых поставщиков **OpenID Connect.**



Istio: OPA

Кастомные действия позволяют интегрировать **Istio** с внешними системами аутентификации, которые реализуют их собственную логику.

Мы можем использовать Open Policy Agent как способ авторизации сервисов и приложений.

Istio: OPA

Kubernetes cluster

Access Management Service

2. validate token, translate to client identity

Service A pods

API client

1. request with access token

Istio Envoy proxy sidecar

4.

3. authorization decision

Service A

OPA sidecar

Service A AuthZ policy mounted

5. delegate to service B

Service B pods

Istio Envoy proxy sidecar

Service A

OPA sidecar

Service B AuthZ policy mounted

K8s + Keycloak

- Единый вход и выход из системы с возможной интеграцией с **Kerberos (LDAP или Active Directory)**
- Поддержка **OpenID Connect** и **SAML 2.0**
- Вход в систему через социальные сети
- Управление учетными записями пользователей как через веб-консоль, так и через **REST API**
- Детальная авторизация для различных сервисов

K8s + Keycloak

ing
Ingress
svc
Service
Keycloak
auth
Pod
Gatekeeper
App
Kubernetes

EKS OIDC

- Более гибкое решение, чем другие провайдеры
- На одну точку отказа меньше (может быть, на несколько меньше)
- Меньшее потребление ресурсов
- Больше модулей на узел

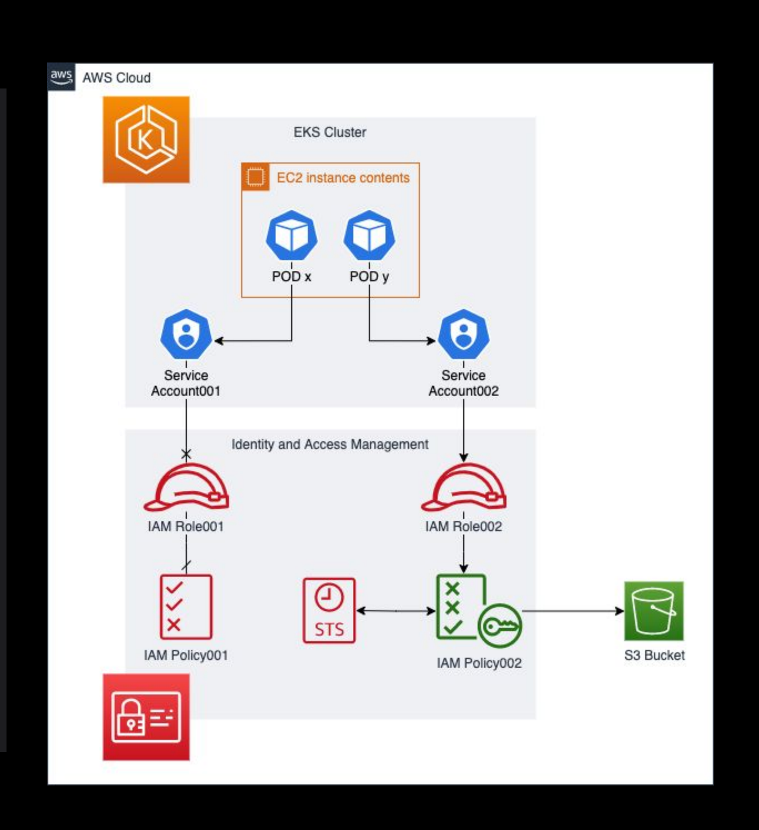


EKS OIDC

- Задержка может сократиться на ~50 мс.

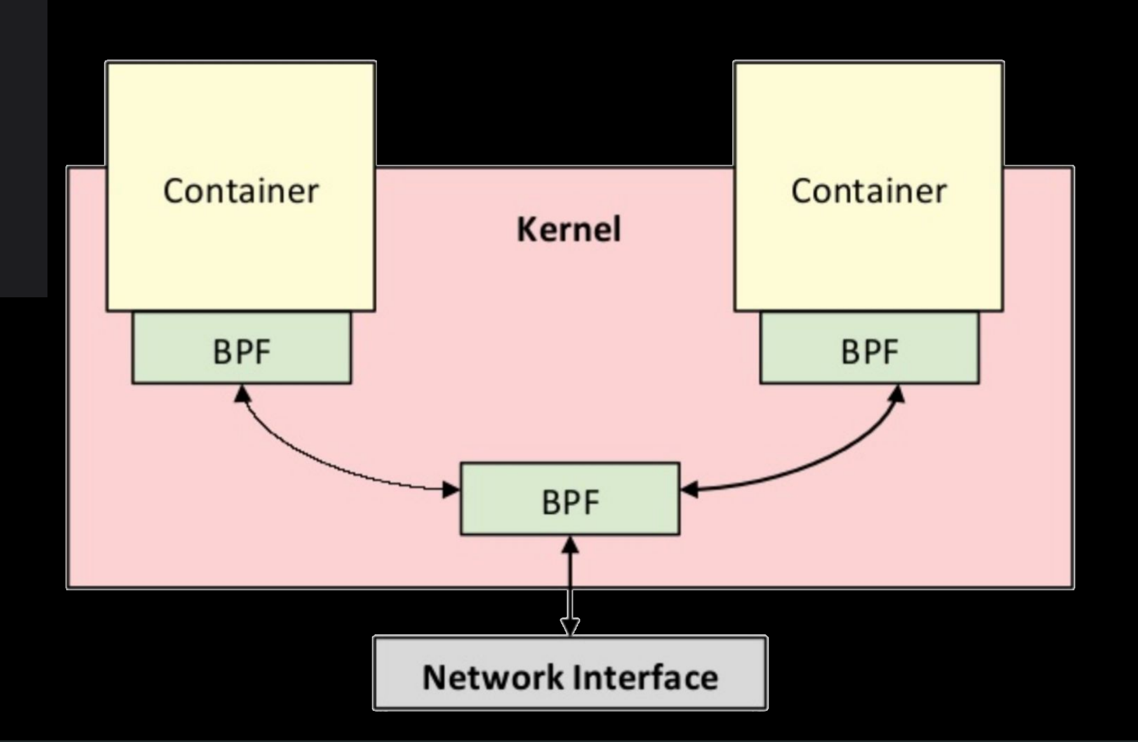
  Особенно для первого запроса
- Предотвращение проблем с

  кэшированием учетных данных
- Лучший аудит
- Легкое внедрение



eBPF: применение в безопасности

- Обнаружение
- Предотвращение
- Мониторинг
- Логирование



eBPF: применение в безопасности

- **Cilium** – сеть на основе eBPF с поддержкой контейнеров, видимостью, безопасность
- **Falco** – инструмент безопасности с открытым исходным кодом для контейнеров, Kubernetes и облаков
- **KubeArmor** – система обеспечения безопасности среды выполнения с поддержкой контейнеров
- **Pixie** – модифицируемая система сбора данных для Kubernetes

Безопасность контейнеров

Безопасность, основанная на функционале eBPF.
Он обнаруживает события, важные для безопасности, такие как:



- запуск процессов
- системные вызовы
- **I/O-активность** (сеть & доступ к файлам)



Tetragon

Логирование





Cluster

fluentbit

fluentbit

fluentbit

fluentd

Elasticsearch

Kibana

influxdb

Grafana

Data Collection

Data Aggregation & Processing

Indexing & storage

Analysis & visualization

K8s-аудит

- KubiScan
- KubeScape
- Kube-bench
- Kube-hunter
- Chekov







Zero Trust-контроли



Контроли и решения

- Целостность логов
- Определите опасные / безопасные действия
- **PKI**
- Обязательный **2FA**
- Привилегированные хосты

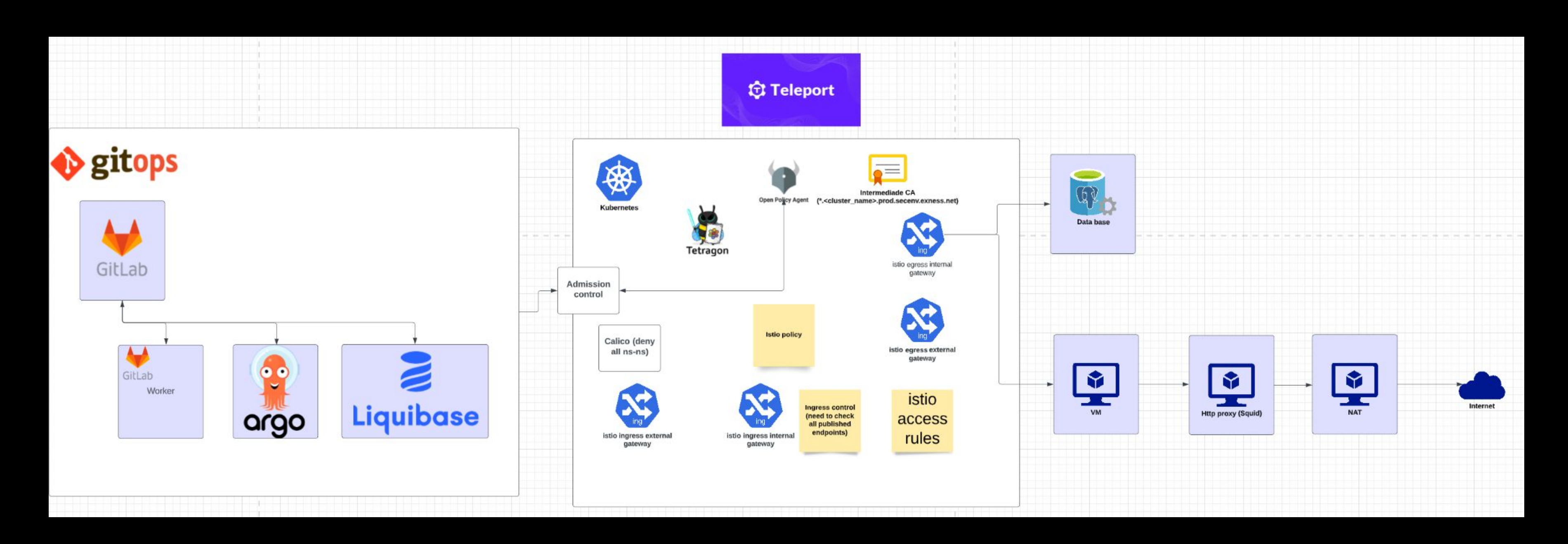
Zero Trust-контроли



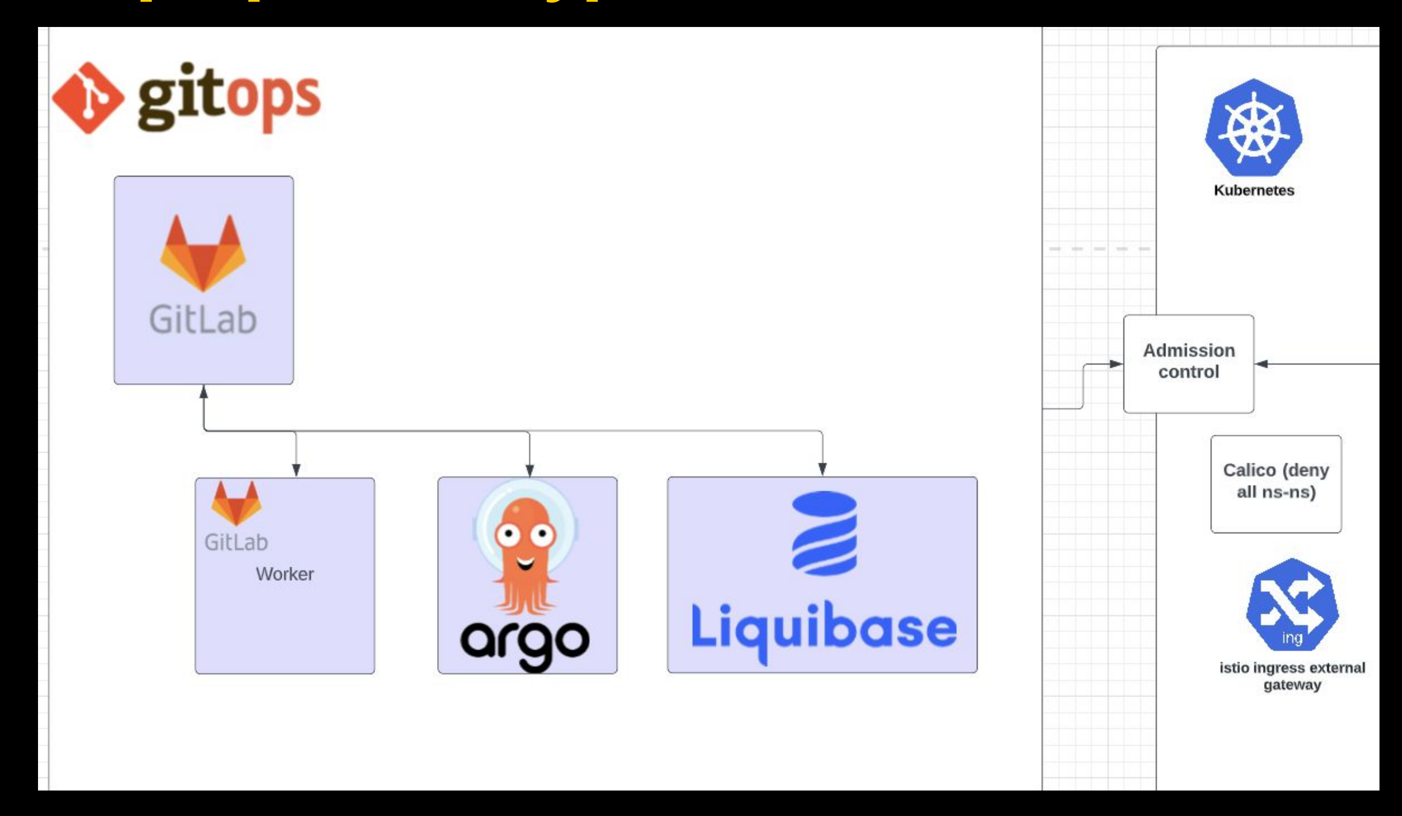
Контроли и решения

- Процесс управления ключами / сертификатами **[все ключи генерируются автоматически]**
- Атестация устройств
- Логирование / Мониторинг / Алертинг / **IR-Плейбуки**
- Проверка действия (**JWT**-подобная подпись события и контекст)
- Политики принудительного исполнения / Гибкая система **mesh** обслуживания с **ACL**

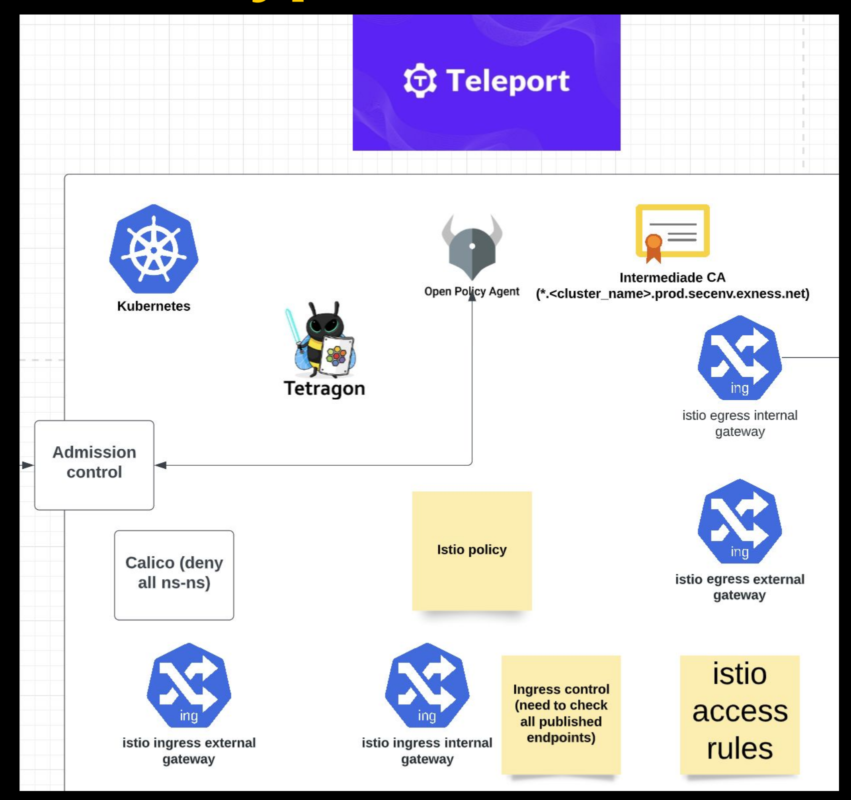
Пример архитектуры



Пример архитектуры



Пример архитектуры



Пример архитектуры

nediade CA
prod.secenv.exness.net)

Data base

ing
istio egress internal
gateway

ing
istio egress external
gateway

istio
access
rules

VM

Http proxy (Squid)

NAT

Internet

GItOPS с согласованиями

Argo CD — это декларативный инструмент непрерывной доставки **GitOps** для **Kubernetes.**



GItOPS с согласованиями

Applications / argocd-bootstrap

0
DAYS ACTIVE

0
DAYS SINCE LAST SYNCHRONIZED

Synced
TO HEAD (ABF1C77)

Healthy

synchronization
SUCCEEDED AT 2019-02-22T13:14:55Z

applications
application

argocd
applica

argocd-bootstrap
4 parameter override(s) application

argocd
appprc

argocd-ops-sync-job
job

argocd-ops-sync-job-t5hs8
completed 0/1 pod

Details
Sync
History
Delete

Предоставление доступа

TELEPORT

Activity

Servers

Applications

Activity

Active Sessions

Session Recordings

Audit Log

Access Requests

Team

Users

Roles

Auth Connectors

Clusters

Access Requests > New Request

Request Role Access

ROLES ALLOWED TO REQUEST

dbadmin ×

REQUEST REASON

check database

SEND REQUEST CANCEL



Спасибо







Александр Сунгуров

Обратная связь
и комментарии по
докладу по ссылке

